

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Horgász Verseny

Készítette: **Koubridis Michael**

Neptunkód: **LWUIJ2**

Dátum: 2024.12.09

Tartalom

Feladat leírása:	3
1 Az adatbázis ER modell.....	4
1.1 Egyedek és az ER modell.....	4
1.2 Adatbázis ER modell	6
1.3 Az XDM modell alapján XML dokumentum készítése	7
1.4 Az XML dokumentum alapján XMLSchema készítése	10
2 DOM program	14
2.1 DOM adatolvasás.....	14
2.2 DOM Adat módosítás	18
2.3 Adatlekérdezés	19
2.4 DOM Adatírás	20

Feladat leírása:

Feladatomban egy olyan adatbázis ER modelljének a megtervezése, ami megfelel a normálformáknak és nem tartalmaz redundanciát. Az adatbázis megfelelő lesz a horgászversenyen történt események tárolására. Eredmények lekérdezésére akár az adott kifogott hal fajtájának kifogott mennyiségére vagy versenyzőre lebontva. Ki mennyit, mit és mikor fogta a halat. Persze látható az adatbázisban, hogy kik vettek részt a versenyen és kik voltak a versenybírók. A hal típus jelöli, hogy milyen típusú a hal például keszeg. Vannak neki tulajdonságai típusa és az indexe ezenfelül, hogy őshonos-e és persze ragadozó vagy növényevő-e. Az index elsődleges kulcs. A kifogott hal egyed tartalmazza a kifogott halat ki mikor és mekkorát fogott. A Versenyző egyed tartalmazza a versenyző adatait, amihez kapcsolódik a díj egyed, ami tartalmazza azokat a tulajdonságokat, hogy melyik versenyző milyen díjakat nyert el. A helyszín egyed, ami kapcsolódik a versenyzőhöz tartalmazza, hogy melyik versenyző melyik horgász helyen ült. A verseny bíró egyed tartalmazza, hogy melyik versenyzőnél volt jelen az adott bíró, és a bírónak is el van tárolva a neve meg a születési adatok.

1 Az adatbázis ER modell

Horgászverseny ER modell tervezését és készítéséhez egy online EE modell készítő felület használtam, aminek a neve ERDPlus. Az ERDPlus-ba könnyen lehet létrehozni egyedeket, és az azokhoz tartalmazó attribútumokat. Ezenfelül könnyen lehet benne ábrázolni az egyedek között lévő kapcsolatokat.

1.1 Egyedek és az ER modell

Hal típus:

- **Index:** A hal típus egyed elsődleges kulcsa.
- **Típusa:** A hal védett hal-e.
- **Őshonos:** Az adott hal-e őshonos.
- **Ragadozó:** A hal ragadozó-e

Kifogott hal:

- **Hazon:** A kifogott hal elsődleges kulcsa.
- **Kg:** A hal súlya.
- **THal:** A hal fajtája

Versenyző:

- **VAzon:** A versenyző egyed elsődleges kulcsa.
- **VNév:** A versenyző neve.
- **Sz_adatok:**
 - **Szul_ido:** A versenyző születési ideje.
 - **Szul_hely:** A versenyző születési helye.
- **Doksi:** A versenyző személyes iratai

Helyszín:

- **Szektor azon:** A helyszín egyed elsődleges kulcsa.
- **Szektor:** A szektor neve
- **Méret:** A szektor mérete.
- **Helyszín típusa:** Magas vagy alacsony part-e.

Verseny Bíró:

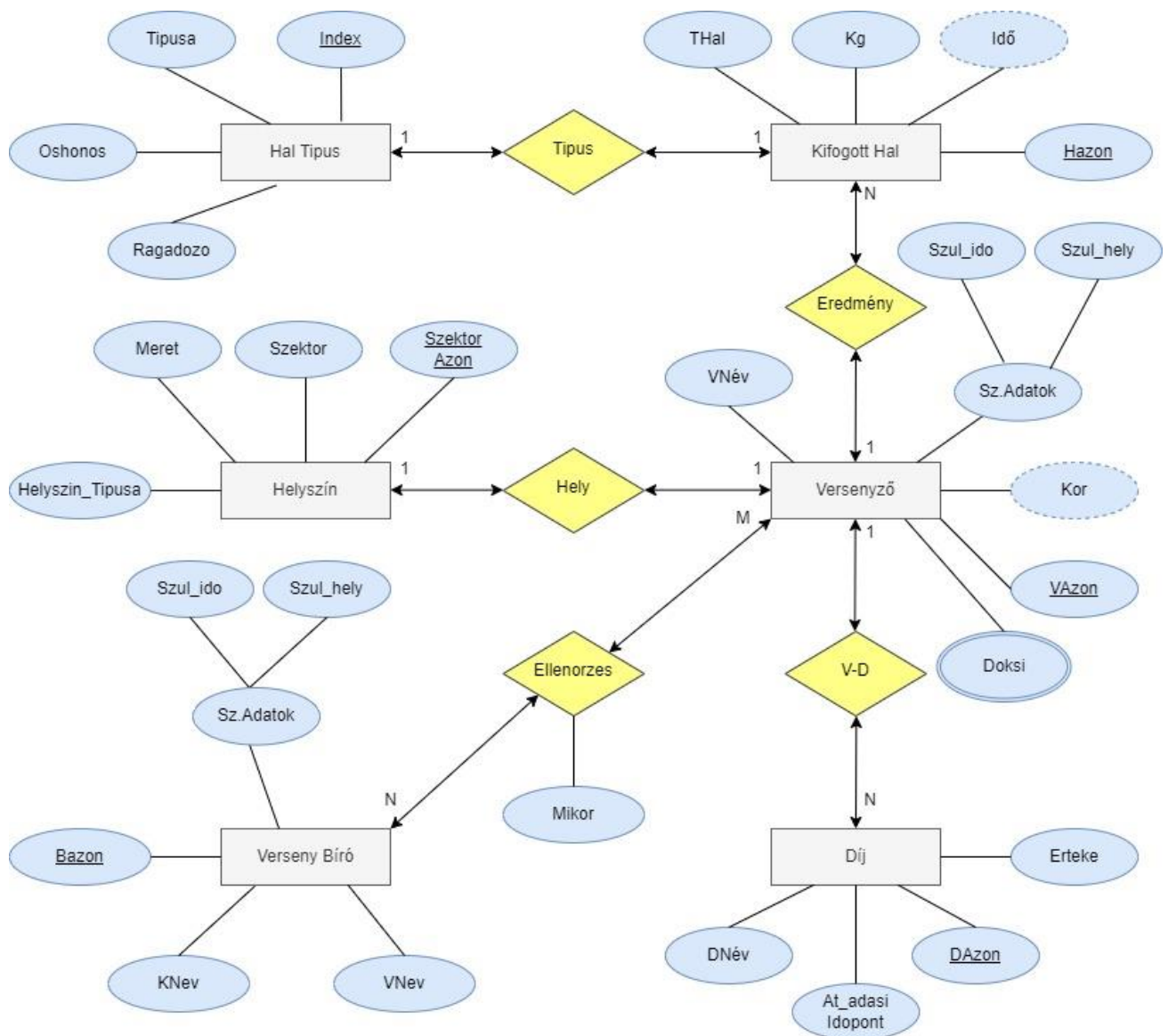
- **Bazon:** A verseny bíró egyed elsődleges kulcsa
- **VNev:** A bíró vezeték neve.
- **Knev:** A bíró keresztnév.

- **Sz_Adatok**

- **Szul_ido:** A bíró születési ideje
- **Szul_hely:** A bíró születési helye

Díj:

- **DAzon:** A díj egyed elsődleges kulcsa
- **DNév:** A díj neve
- **At_adasi idopont:** Mikor lett átadva a díj
- **Erteke:** A díjnak az értéke



1. ábra: A horgászverseny ER modellje

1.3 Az XDM modell alapján XML dokumentum készítése

```
<?xml version="1.0" encoding="UTF-8" ?>
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaLWUIJ2.xsd">
  <Versenyzok>
    <Versenyzo VAzon="1" Szektorazon="1" Dazon="1">
      <VNev>Bence</VNev>
      <Sz_adatok>
        <Szul_ido>1997.01.06</Szul_ido>
        <Szul_hely>Gyula</Szul_hely>
      </Sz_adatok>
    </Versenyzo>
    <Versenyzo VAzon="2" Szektorazon="2" Dazon="2">
      <VNev>Akos</VNev>
      <Sz_adatok>
        <Szul_ido>2000.10.15</Szul_ido>
        <Szul_hely>Miskolc</Szul_hely>
      </Sz_adatok>
    </Versenyzo>
    <Versenyzo VAzon="3" Szektorazon="3" Dazon="3">
      <VNev>Alex</VNev>
      <Sz_adatok>
        <Szul_ido>1999.03.20</Szul_ido>
        <Szul_hely>Budapest</Szul_hely>
      </Sz_adatok>
    </Versenyzo>
  </Versenyzok>
  <Doksik>
    <Doksi Vazon="1">
      <Igazolvany>Taj_Kartya</Igazolvany>
    </Doksi>
    <Doksi Vazon="2">
      <Igazolvany>Vezetoi_Engedely</Igazolvany>
    </Doksi>
    <Doksi Vazon="3">
      <Igazolvany>Szig_szam</Igazolvany>
    </Doksi>
    <Doksi Vazon="4">
      <Igazolvany>Lakcim_kartya</Igazolvany>
    </Doksi>
  </Doksik>
  <Verseny_Biro>
    <Verseny_Biro BAzon="1">
      <VNev>Kiss</VNev>
      <KNev>Joe</KNev>
      <Sz_adatok>
        <Szul_ido>1980.06.20</Szul_ido>
        <Szul_hely>Budapest</Szul_hely>
      </Sz_adatok>
    </Verseny_Biro>
    <Verseny_Biro BAzon="2">
      <VNev>Nagy</VNev>
      <KNev>Ferenc</KNev>
      <Sz_adatok>
        <Szul_ido>1985.12.03</Szul_ido>
        <Szul_hely>Sopron</Szul_hely>
      </Sz_adatok>
    </Verseny_Biro>
    <Verseny_Biro BAzon="3">
      <VNev>Kovacs</VNev>
```

```

        <KNev>Janos</KNev>
        <Sz_adatok>
            <Szul_ido>1975.02.10</Szul_ido>
            <Szul_hely>Gyor</Szul_hely>
        </Sz_adatok>
    </Verseny_Biro>
</Verseny_Birok>
<Kifogott_Halak>
    <Kifogott_hal Hazon="1" Halindex="1" VerAzon="1">
        <THal>Csuka</THal>
        <Kg>5</Kg>
        <ido>14:00</ido>
    </Kifogott_hal>
    <Kifogott_hal Hazon="2" Halindex="2" VerAzon="2">
        <THal>Ponty</THal>
        <Kg>10</Kg>
        <ido>15:00</ido>
    </Kifogott_hal>
    <Kifogott_hal Hazon="3" Halindex="3" VerAzon="3">
        <THal>Amur</THal>
        <Kg>8</Kg>
        <ido>17:00</ido>
    </Kifogott_hal>
</Kifogott_Halak>
<Ellenorzesek>
    <Ellenorzes VersenyzoId="2" VBiroId="1">
        <Mikor>17:55</Mikor>
    </Ellenorzes>
    <Ellenorzes VersenyzoId="1" VBiroId="1">
        <Mikor>17:50</Mikor>
    </Ellenorzes>
    <Ellenorzes VersenyzoId="3" VBiroId="3">
        <Mikor>17:45</Mikor>
    </Ellenorzes>
    <Ellenorzes VersenyzoId="2" VBiroId="3">
        <Mikor>17:40</Mikor>
    </Ellenorzes>
    <Ellenorzes VersenyzoId="1" VBiroId="2">
        <Mikor>17:35</Mikor>
    </Ellenorzes>
</Ellenorzesek>
<Dijak>
    <Dij Dazon="1">
        <DNev>Legtobb_Hal</DNev>
        <Erteke>5000</Erteke>
        <At_adasi_idopont>18:00</At_adasi_idopont>
    </Dij>
    <Dij Dazon="2">
        <DNev>Legnagyobb_Hal</DNev>
        <Erteke>5000</Erteke>
        <At_adasi_idopont>18:30</At_adasi_idopont>
    </Dij>
    <Dij Dazon="3">
        <DNev>Gyoztes</DNev>
        <Erteke>20000</Erteke>
        <At_adasi_idopont>19:00</At_adasi_idopont>
    </Dij>
</Dijak>
<Helyszinek>
    <Helyszin SzektorAzon="1">
        <Helyszin_tipusa>Alacsony_part</Helyszin_tipusa>
    </Helyszin>
</Helyszinek>

```



```

        <Meret>2033</Meret>
        <Szektor>A</Szektor>
    </Helyszin>
    <Helyszin SzektorAzon="2">
        <Helyszin_tipusa>Magas_part</Helyszin_tipusa>
        <Meret>25</Meret>
        <Szektor>B</Szektor>
    </Helyszin>
    <Helyszin SzektorAzon="3">
        <Helyszin_tipusa>Alacsony_part</Helyszin_tipusa>
        <Meret>30</Meret>
        <Szektor>C</Szektor>
    </Helyszin>
</Helyszinek>
<Hal_Tipusok>
    <Hal_Tipus index="1">
        <Ragadozo>igen</Ragadozo>
        <Tipusa>VedettHal</Tipusa>
        <Oshonos>nem</Oshonos>
    </Hal_Tipus>
    <Hal_Tipus index="2">
        <Ragadozo>nem</Ragadozo>
        <Tipusa>VedettHal</Tipusa>
        <Oshonos>igen</Oshonos>
    </Hal_Tipus>
    <Hal_Tipus index="3">
        <Ragadozo>igen</Ragadozo>
        <Tipusa>NemVedettHal</Tipusa>
        <Oshonos>igen</Oshonos>
    </Hal_Tipus>
</Hal_Tipusok>
</root>

```

1.4 Az XML dokumentum alapján XMLSchema készítése

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Versenyzo" maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="VNev"
type="xs:string" />
              <xs:element name="Sz_adatok">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element
name="Szul_ido" type="szulido" />
                    <xs:element
name="Szul_hely" type="xs:string" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="VAzon"
type="xs:string" />
            <xs:attribute name="Szektorazon"
type="xs:string" />
            <xs:attribute name="Dazon"
type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Doksik">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Doksi" maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Igazolvany"
type="xs:string" />
            </xs:sequence>
            <xs:attribute name="Vazon"
type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Verseny_Birok">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Verseny_Biro" minOccurs="1"
maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="VNev"
```

```

type="xs:string" />
type="xs:string" />
name="Szul_ido" type="szulido" />
name="Szul_hely" type="xs:string" />
type="xs:string"/>
maxOccurs="1000">
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="xs:string"/>
type="idoType"/>
type="xs:string"/>
type="xs:string"/>

```

```

<xs:element name="KNev">
  <xs:element name="Sz_adatok">
    <xs:complexType>
      <xs:sequence>
        <xs:element
          <xs:element
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="BAzon"
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Kifogott_Halak">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Kifogott_hal" minOccurs="1"
    <xs:complexType>
      <xs:sequence>
        <xs:element name="THal"
          <xs:element name="Kg"
            <xs:element name="ido"
          </xs:sequence>
          <xs:attribute name="Hazon"
            <xs:attribute name="Halindex"
              <xs:attribute name="VerAzon"
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Ellenorzesek">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Ellenorzes" minOccurs="1"
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Mikor"
          </xs:sequence>
          <xs:attribute name="VersenyzoId"
            <xs:attribute name="VBiroId"
          </xs:complexType>

```

```

        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Dijak">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Dij" minOccurs="1"
maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="DNev"
type="xs:string"/>
              <xs:element name="Erteke"
type="xs:string"/>
              <xs:element name="At_adasi_idopont"
type="idoType"/>
            </xs:sequence>
            <xs:attribute name="Dazon"
type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Helyszinek">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Helyszin" minOccurs="1"
maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Helyszin_tipusa"
type="xs:string"/>
              <xs:element name="Meret"
type="xs:positiveInteger"/>
              <xs:element name="Szektor"
type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="SzektorAzon"
type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Hal_Tipusok">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Hal_Tipus" minOccurs="1"
maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Ragadozo"
type="xs:string"/>
              <xs:element name="Tipusa"
type="xs:string"/>
              <xs:element name="Oshonos"
type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="index"

```

```

type="xs:string"/>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="di_j_Key">
  <xs:selector xpath="Dij"/>
  <xs:field xpath="@Dazon"/>
</xs:key>
<xs:key name="Versenyzo_Key">
  <xs:selector xpath="Versenyzo"/>
  <xs:field xpath="@VAzon"/>
</xs:key>
<xs:key name="kifogottHal_Key">
  <xs:selector xpath="Kifogott_hal"/>
  <xs:field xpath="@Hazon"/>
</xs:key>
<xs:key name="VersenyBir_Key">
  <xs:selector xpath="Verseny_Biro"/>
  <xs:field xpath="@Bazon"/>
</xs:key>
<xs:key name="Hal_Tipus_key">
  <xs:selector xpath="Hal_Tipus"/>
  <xs:field xpath="@index"/>
</xs:key>
<xs:key name="helyszin_key">
  <xs:selector xpath="Helyszin"/>
  <xs:field xpath="@SzektorAzon"/>
</xs:key>
<xs:keyref name="Fkey_dij_Versenyzo" refer="di_j_Key">
  <xs:selector xpath="Versenyzo"/>
  <xs:field xpath="@Dazon"/>
</xs:keyref>
<xs:keyref name="Fkey_H-V_VersenyBiro" refer="VersenyBir_Key">
  <xs:selector xpath="Ellenorzes"/>
  <xs:field xpath="@VBiroId"/>
</xs:keyref>
<xs:keyref name="Fkey_H-V_Versenyzo" refer="Versenyzo_Key">
  <xs:selector xpath="Ellenorzes"/>
  <xs:field xpath="@VersenyzoId"/>
</xs:keyref>
<xs:keyref name="Fkey_Hal_Tipus_Kifogott_hal"
refer="Hal_Tipus_key">
  <xs:selector xpath="Kifogott_hal"/>
  <xs:field xpath="@Halindex"/>
</xs:keyref>
<xs:keyref name="Fkey_Kifogott_hal_Versenyzo"
refer="Versenyzo_Key">
  <xs:selector xpath="Kifogott_hal"/>
  <xs:field xpath="@VerAzon"/>
</xs:keyref>
<xs:keyref name="Fkey_Helyszin_Versenyzo" refer="helyszin_key">
  <xs:selector xpath="Versenyzo"/>
  <xs:field xpath="@Szektorazon"/>
</xs:keyref>
<xs:keyref name="Fkey_Doksi_Versenyzo" refer="Versenyzo_Key">
  <xs:selector xpath="Doksi"/>
  <xs:field xpath="@Vazon"/>

```

```

        </xs:keyref>
    </xs:element>
    <xs:simpleType name="idoType">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{1}[0-9]{1}:[0-9]{1}[0-9]{1}" />
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="szulido">
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{4}[.]\d{2}[.]\d{2}" />
            <xs:length value="10" />
        </xs:restriction>
    </xs:simpleType>
</xs:schema>

```

2 DOM program

A DOM programot Java környezetben készítettem el. A DOM program tartalmaz adat módosítást, adat lekérdezést, illetve adatolvasást is.

2.1 DOM adatolvasás

Dom segítségével ki listázza az összes adatott.

```

package hu.domparsing.LWUIJ2;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DomReadLWUIJ2 {
    public static void main(String[] args) {
        NodeList list;
        try {
            DocumentBuilderFactory
factory=DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder=factory.newDocumentBuilder();
            //fájl beolvasása
            Document document=documentBuilder.parse(new File("XMLLWUIJ2.xml"));
            document.getDocumentElement().normalize();
            //Gyökér elem megkeresése
            System.out.println("Root element : " +
document.getDocumentElement().getNodeName());
            System.out.println("-----");
            //Aktuális elem meghatározása
            list=document.getElementsByTagName("Versenyző");

            for (int i=0;i<list.getLength();i++) {
                Node node=list.item(i);
                System.out.println("\nAktuális elem: " +

```

```

node.getNodeName());
    //Versenyzok adatainak kiirása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Versenyző id: " +
element.getAttribute("VAzon"));
        System.out.println("Szektorazonosító id: " +
element.getAttribute("Szektorazon"));
        System.out.println("Díj id: " +
element.getAttribute("Dazon"));
        System.out.println("VersenyzőNév: "
+
element.getElementsByTagName("VNev").item(0).getTextContent());

        //Versenyző születési adatainak kiirása
        Node nodeszuladat = list.item(i);
        if (nodeszuladat.getNodeType() == Node.ELEMENT_NODE) {
            Element elementszuldat = (Element) node;
            System.out.println("Születési idő: " +
elementszuldat.getElementsByTagName("Szul_ido").item(0).getTextContent());
            System.out.println("Születési hely: " +
elementszuldat.getElementsByTagName("Szul_hely").item(0).getTextContent());
        }
    }
}
//Aktuális elem meghatározása
list = document.getElementsByTagName("Doksi");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiirása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Dokumentum id: " +
element.getAttribute("Vazon"));
        System.out.println("Igazolvány: " +
element.getElementsByTagName("Igazolvány").item(0).getTextContent());
    }
}
//Aktuális elem meghatározása
list = document.getElementsByTagName("Verseny_Biro");

for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Verseny Bírok adatainak kiirása
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("Verseny Bíró id: " +
element.getAttribute("BAzon"));
        System.out.println("Vezetéknév: " +
element.getElementsByTagName("VNev").item(0).getTextContent());
        System.out.println("Keresztnév: " +
element.getElementsByTagName("KNev").item(0).getTextContent());

        //Bíró születési adatainak kiirása
        Node nodeszuladat = list.item(i);
        if (nodeszuladat.getNodeType() == Node.ELEMENT_NODE) {
            Element elementszuldat = (Element) node;
            System.out.println("Születési idő: " +

```

```

elementsuldat.getElementsByTagName("Szul_ido").item(0).getTextContent());
        System.out.println("Születési hely: " +
elementsuldat.getElementsByTagName("Szul_hely").item(0).getTextContent());
    }
}
//Aktuális elem meghatározása
list=document.getElementsByTagName("Kifogott_hal");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Hal id: " +
element.getAttribute("Vazon"));
        System.out.println("KifogottHalnak az idje: " +
element.getAttribute("Halindex"));
        System.out.println("Versenyző idje: " +
element.getAttribute("VerAzon"));
        System.out.println("Tipusa: " +
element.getElementsByTagName("THal").item(0).getTextContent());
        System.out.println("Mérete: " +
element.getElementsByTagName("Kg").item(0).getTextContent());
        System.out.println("Mikor fogták: " +
element.getElementsByTagName("ido").item(0).getTextContent());

    }
}
//Aktuális elem meghatározása
list=document.getElementsByTagName("Ellenorzes");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Biro id: " +
element.getAttribute("VBiroId"));
        System.out.println("Versenyző idje: " +
element.getAttribute("VersenyzoId"));
        System.out.println("Mikor: " +
element.getElementsByTagName("Mikor").item(0).getTextContent());
    }
}
//Aktuális elem meghatározása
list=document.getElementsByTagName("Dij");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Dij id: " +
element.getAttribute("VBiroId"));
        System.out.println("Értéke: " +
element.getElementsByTagName("Erteke").item(0).getTextContent());
        System.out.println("Át adási időpont: " +

```



```

element.getElementsByTagName("At_adasi_idopont").item(0).getTextContent());
    }
}

//Aktuális elem meghatározása
list=document.getElementsByTagName("Helyszin");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Szektor id: " +
element.getAttribute("SzektorAzon"));
        System.out.println("Helyszin tipusa: " +
element.getElementsByTagName("Helyszin_tipusa").item(0).getTextContent());
        System.out.println("Mérete: " +
element.getElementsByTagName("Meret").item(0).getTextContent());
        System.out.println("Szektor: " +
element.getElementsByTagName("Szektor").item(0).getTextContent());
    }
}

//Aktuális elem meghatározása
list=document.getElementsByTagName("Hal_Tipus");

for(int i=0;i<list.getLength();i++){
    Node node=list.item(i);
    System.out.println("\nAktuális elem: " + node.getNodeName());
    //Doksi adatainak kiírása
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element=(Element) node;
        System.out.println("Hal_tipus id: " +
element.getAttribute("index"));
        System.out.println("Ragadozó: " +
element.getElementsByTagName("Ragadozo").item(0).getTextContent());
        System.out.println("Tipusa: " +
element.getElementsByTagName("Tipusa").item(0).getTextContent());
        System.out.println("Oshonos: " +
element.getElementsByTagName("Oshonos").item(0).getTextContent());
    }
}

    }catch (ParserConfigurationException e){
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

2.2 DOM Adat módosítás

A Versenyző táblából megkeresi azt a versenyzőt, akinek az id-je 2-es majd a születési helyét módosítja a kódban megadottra majd egy új xml fájlba kiírja a módosított xml fájlt.

```
package hu.domparsing.LWUIJ2;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Result;
import javax.xml.transform.Source;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;

public class DOMModifyLWUIJ2 {
    public static void main(String[] args) {
        NodeList nodeList;
        try{
            DocumentBuilderFactory factory
=DocumentBuilderFactory.newInstance();
            DocumentBuilder builder=factory.newDocumentBuilder();
            //Fájl beolvasása
            Document document=builder.parse(new File("XMLLWUIJ2.xml"));
            document.getDocumentElement().normalize();
            //Aktuális elem meghatározása
            nodeList=document.getElementsByTagName("Versenyzo");
            for(int i =0;i<nodeList.getLength();i++){
                Node node=nodeList.item(i);
                System.out.println("\nAktuális elem: " +
node.getNodeName());
                //Versenyzo adatainak kiírása
                if(node.getNodeType()==Node.ELEMENT_NODE){
                    Element element=(Element) node;
                    //Versenyzo idjének eltárolása
                    String id= element.getAttribute("VAzon");

                    System.out.println("Versenyzo id: " +
element.getAttribute("VAzon"));
                    System.out.println("Szektorazonosító id: " +
element.getAttribute("Szektorazon"));
                    System.out.println("Díj id: " +
element.getAttribute("Dazon"));
                    System.out.println("VersenyzőNév: "
+
element.getElementsByTagName("VNev").item(0).getTextContent());

                    //Versenyző születési adatainak kiírása
                    Node nodeszuladat=nodeList.item(i);
                    if(nodeszuladat.getNodeType()==Node.ELEMENT_NODE){
                        Element elementszuldat=(Element) node;
```

```

        //id vizsgálata ha megegyezik a megadot id-vel
        akkor születési hely módosítás
        if(id.equals("2")) {

            elementszuldat.getElementsByTagName("Szul_hely").item(0).setTextContent("Los Angeles");

        }
        System.out.println("Születési idő: " +
            elementszuldat.getElementsByTagName("Szul_ido").item(0).getTextContent());
        System.out.println("Születési hely: " +
            elementszuldat.getElementsByTagName("Szul_hely").item(0).getTextContent());
    }
}

//létrehozza az xml fájlt módosítva
Transformer transformer=
TransformerFactory.newInstance().newTransformer();
Source input=new DOMSource(document);
Result output=new StreamResult(new
File("XMLLWUIJ2Modify.xml"));
System.out.println("létrejött az új file XMLLWUIJ2Modify.xml
néven");

    transformer.transform(input,output);
} catch (ParserConfigurationException e){
    e.printStackTrace();
} catch (Exception e){
    e.printStackTrace();
}
}
}
}

```

2.3 Adatlekérdezés

A hal típus táblából kiírja ki listázza azokat, amelyek védettek, és kiírja a consolera.

```

package hu.domparse.LWUIJ2;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.File;
import java.io.IOException;

public class DOMQueryLWUIJ2 {
    public static void main(String[] args) {
        try {
            NodeList nodeList;

            DocumentBuilderFactory factory
=DocumentBuilderFactory.newInstance();
            DocumentBuilder builder=factory.newDocumentBuilder();
            //Fájl beolvasása
            Document document=builder.parse(new File("XMLLWUIJ2.xml"));
            document.getDocumentElement().normalize();

```

```

//Aktuális elem meghatározása
nodeList=document.getElementsByTagName("Hal_Tipus");

for(int i =0;i<nodeList.getLength();i++){
    Node node=nodeList.item(i);

    //Hal tipusok adatainak kiírása
    if(node.getNodeType()==Node.ELEMENT_NODE){
        Element element=(Element) node;
        String tipus=
element.getElementsByTagName("Tipusa").item(0).getTextContent();

        if(tipus.equals("VedettHal"))
        {
            System.out.println("\nAktuális elem: " +
node.getNodeName());
            System.out.println("-----");
            System.out.println("");
            System.out.println("Hal tipus id: " +
element.getAttribute("index"));
            System.out.println("Ragadozó: " +
element.getElementsByTagName("Ragadozo").item(0).getTextContent());
            System.out.println("Tipusa: " +
element.getElementsByTagName("Tipusa").item(0).getTextContent());
            System.out.println("Oshonos: " +
element.getElementsByTagName("Oshonos").item(0).getTextContent());

        }
    }
}
} catch (ParserConfigurationException | IOException | SAXException
e){
    e.printStackTrace();
}
}
}

```

2.4 DOM Adatírás

Az xml fájlhoz hozzá adunk új értékeket majd ezt egy új xml fájlba kiírjuk.

```
package hu.domparsing.LWUIJ2;
```

```

import org.w3c.dom.*;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.*;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;

public class DOMWriteLWUIJ2 {

    public static void main(String[] args) {

        try {
            DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();

```

```

        Document document = builder.newDocument();

        Element root = document.createElement("root");
        root.setAttribute("xmlns:xsi",
"http://www.w3.org/2001/XMLSchema-instance");
        root.setAttribute("xsi:noNamespaceSchemaLocation",
"DOMParseLWUIJ2/XMLSchemaLWUIJ2.xsd");
        document.appendChild(root);

        Element versenyzok = document.createElement("Versenyzok");
        root.appendChild(versenyzok);
        addVersenyzok(document, versenyzok);

        Element doksik = document.createElement("Doksik");
        root.appendChild(doksik);
        addDoksik(document, doksik);

        Element versenyBirok = document.createElement("Verseny_Birok");
        root.appendChild(versenyBirok);
        addVersenyBirok(document, versenyBirok);

        Element kifogottHalak =
document.createElement("Kifogott_Halak");
        root.appendChild(kifogottHalak);
        addKifogottHalak(document, kifogottHalak);
        Element ellenorzesek = document.createElement("Ellenorzesek");
        root.appendChild(ellenorzesek);

        addEllenorzesek(document, ellenorzesek);
        Element dijak = document.createElement("Dijak");
        root.appendChild(dijak);
        addDijak(document, dijak);

        Element helyszinek = document.createElement("Helyszinek");
        root.appendChild(helyszinek);
        addHelyszinek(document, helyszinek);

        Element halTipusok = document.createElement("Hal_Tipusok");
        root.appendChild(halTipusok);
        addHalTipusok(document, halTipusok);

        TransformerFactory transformerFactory =
TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        transformer.setOutputProperty(OutputKeys.INDENT, "yes");
        DOMSource domSource = new DOMSource(document);
        StreamResult streamResult = new StreamResult(new
File("XMLLWUIJ2out.xml"));
        transformer.transform(domSource, streamResult);
        System.out.println("Az XML fájl sikeresen létrehozva!");

    } catch (ParserConfigurationException | TransformerException e) {
        e.printStackTrace();
    }
}

private static void addVersenyzok(Document document, Element

```

```

versenyzo) {

    String[][] versenyzoAdatok = {
        {"1", "1", "1", "Bence", "1997.01.06", "Gyula"},
        {"2", "2", "2", "Akos", "2000.10.15", "Miskolc"},
        {"3", "3", "3", "Alex", "1999.03.20", "Budapest"}
    };

    for (String[] adat : versenyzoAdatok) {
        Element versenyzo = document.createElement("Versenyzo");
        versenyzo.setAttribute("VAzon", adat[0]);
        versenyzo.setAttribute("Szektorazon", adat[1]);
        versenyzo.setAttribute("Dazon", adat[2]);
        versenyzo.appendChild(versenyzo);

        Element vNev = document.createElement("VNev");
        vNev.appendChild(document.createTextNode(adat[3]));
        versenyzo.appendChild(vNev);

        Element szAdatok = document.createElement("Sz_adatok");
        versenyzo.appendChild(szAdatok);

        Element szulIdo = document.createElement("Szul_ido");
        szulIdo.appendChild(document.createTextNode(adat[4]));
        szAdatok.appendChild(szulIdo);

        Element szulHely = document.createElement("Szul_hely");
        szulHely.appendChild(document.createTextNode(adat[5]));
        szAdatok.appendChild(szulHely);

    }
}

private static void addDoksik(Document document, Element doksik) {
    String[] igazolvanyok = {"Taj_Kartya", "Vezetoi_Engedely",
        "Szig_szam", "Lakcim_kartya"};

    for (int i = 1; i <= igazolvanyok.length; i++) {
        Element doksi = document.createElement("Doksi");
        doksi.setAttribute("Vazon", String.valueOf(i));
        doksik.appendChild(doksi);

        Element igazolvany = document.createElement("Igazolvany");
        igazolvany.appendChild(document.createTextNode(igazolvanyok[i -
1])));
        doksi.appendChild(igazolvany);
    }
}

private static void addVersenyBirok(Document document, Element
versenyBirok) {
    String[][] birok = {
        {"1", "Kiss", "Joe", "1980.06.20", "Budapest"},
        {"2", "Nagy", "Ferenc", "1985.12.03", "Sopron"},
        {"3", "Kovacs", "Janos", "1975.02.10", "Gyor"}
    };
}

```

```

        for (String[] biro : birok) {
            Element versenyBiro = document.createElement("Verseny_Biro");
            versenyBiro.setAttribute("BAzon", biro[0]);
            versenyBirok.appendChild(versenyBiro);

            Element vNev = document.createElement("VNev");
            vNev.appendChild(document.createTextNode(biro[1]));
            versenyBiro.appendChild(vNev);

            Element kNev = document.createElement("KNev");
            kNev.appendChild(document.createTextNode(biro[2]));
            versenyBiro.appendChild(kNev);

            Element szAdatok = document.createElement("Sz_adatok");
            versenyBiro.appendChild(szAdatok);
            Element szulIdo = document.createElement("Szul_ido");
            szulIdo.appendChild(document.createTextNode(biro[3]));
            szAdatok.appendChild(szulIdo);

            Element szulHely = document.createElement("Szul_hely");
            szulHely.appendChild(document.createTextNode(biro[4]));
            szAdatok.appendChild(szulHely);
        }
    }

    private static void addKifogottHalak(Document document, Element
kifogottHalak) {
        String[][] halak = {
            {"1", "1", "1", "Csuka", "5", "14:00"},
            {"2", "2", "2", "Ponty", "10", "15:00"},
            {"3", "3", "3", "Amur", "8", "17:00"}
        };

        for (String[] hal : halak) {
            Element kifogottHal = document.createElement("Kifogott_hal");
            kifogottHal.setAttribute("Hazon", hal[0]);
            kifogottHal.setAttribute("Halindex", hal[1]);
            kifogottHal.setAttribute("VerAzon", hal[2]);
            kifogottHalak.appendChild(kifogottHal);

            Element tHal = document.createElement("THal");
            tHal.appendChild(document.createTextNode(hal[3]));
            kifogottHal.appendChild(tHal);

            Element kg = document.createElement("Kg");
            kg.appendChild(document.createTextNode(hal[4]));
            kifogottHal.appendChild(kg);

            Element ido = document.createElement("ido");
            ido.appendChild(document.createTextNode(hal[5]));
            kifogottHal.appendChild(ido);
        }
    }

    private static void addEllenorzesek(Document document, Element
ellenorzesek) {
        String[][] ellenorzesekAdatok = {

```

```

        {"2", "1", "17:55"},
        {"1", "1", "17:50"},
        {"3", "3", "17:45"},
        {"2", "3", "17:40"},
        {"1", "2", "17:35"}
    };

    for (String[] adat : ellenorzesekAdatok) {
        Element ellenorzes = document.createElement("Ellenorzes");
        ellenorzes.setAttribute("VersenyzoId", adat[0]);
        ellenorzes.setAttribute("VBiroId", adat[1]);
        ellenorzesek.appendChild(ellenorzes);

        Element mikor = document.createElement("Mikor");
        mikor.appendChild(document.createTextNode(adat[2]));
        ellenorzes.appendChild(mikor);
    }
}

private static void addDijak(Document document, Element dijak) {
    String[][] dijakAdatok = {
        {"1", "Legtobb_Hal", "5000", "18:00"},
        {"2", "Legnagyobb_Hal", "5000", "18:30"},
        {"3", "Gyoztes", "20000", "19:00"}
    };

    for (String[] adat : dijakAdatok) {
        Element dij = document.createElement("Dij");
        dij.setAttribute("Dazon", adat[0]);
        dijak.appendChild(dij);

        Element dNev = document.createElement("DNev");
        dNev.appendChild(document.createTextNode(adat[1]));
        dij.appendChild(dNev);

        Element erteke = document.createElement("Erteke");
        erteke.appendChild(document.createTextNode(adat[2]));
        dij.appendChild(erteke);

        Element atadasiIdo =
document.createElement("At_adasi_idopont");
        atadasiIdo.appendChild(document.createTextNode(adat[3]));
        dij.appendChild(atadasiIdo);
    }
}

private static void addHelyszinek(Document document, Element
helyszinek) {
    String[][] helyszinekAdatok = {
        {"1", "Alacsony_part", "2033", "A"},
        {"2", "Magas_part", "25", "B"},
        {"3", "Alacsony_part", "30", "C"}
    };

    for (String[] adat : helyszinekAdatok) {
        Element helyszin = document.createElement("Helyszin");
        helyszin.setAttribute("SzektorAzon", adat[0]);
        helyszinek.appendChild(helyszin);
    }
}

```



```

        Element helyszinTipusa =
document.createElement("Helyszin_tipusa");
        helyszinTipusa.appendChild(document.createTextNode(adat[1]));
        helyszin.appendChild(helyszinTipusa);

        Element meret = document.createElement("Meret");
        meret.appendChild(document.createTextNode(adat[2]));
        helyszin.appendChild(meret);

        Element szektor = document.createElement("Szektor");
        szektor.appendChild(document.createTextNode(adat[3]));
        helyszin.appendChild(sektor);
    }
}

private static void addHalTipusok(Document document, Element
halTipusok) {
    String[][] halTipusAdatok = {
        {"1", "igen", "VedettHal", "nem"},
        {"2", "nem", "VedettHal", "igen"},
        {"3", "igen", "NemVedettHal", "igen"}
    };

    for (String[] adat : halTipusAdatok) {
        Element halTipus = document.createElement("Hal_Tipus");
        halTipus.setAttribute("index", adat[0]);
        halTipusok.appendChild(halTipus);

        Element ragadozo = document.createElement("Ragadozo");
        ragadozo.appendChild(document.createTextNode(adat[1]));
        halTipus.appendChild(ragadozo);

        Element tipusa = document.createElement("Tipusa");
        tipusa.appendChild(document.createTextNode(adat[2]));
        halTipus.appendChild(tipusa);

        Element oshonos = document.createElement("Oshonos");
        oshonos.appendChild(document.createTextNode(adat[3]));
        halTipus.appendChild(oshonos);
    }
}
}

```